EXHIBIT 47

FILED UNDER SEAL

New Building AI Agents is Now Easy →

Docs Get Support Contact Sales

docker AI Products Developers Pricing Support Blog Company Sign In Get Started

Introducing runC: a lightweight universal container runtime

Solomon Hykes

Posted

Jun 22, 2015

Post Tags

- Go
- Linux
- runc
- runtime

Post Categories

- Community

Spinning Out Docker's Plumbing: Part 1: Introducing runC

On Infrastructure Plumbing

To build a platform like Docker you need a lot of infrastructure plumbing; in fact over the past two years even though our code base has grown to tens of thousands of lines of code; roughly 50% of it is plumbing! Infrastructure plumbing is made of small software tools which perform basic fundamental tasks in the most reliable and simple way possible. It is invisible and under-appreciated especially given that plumbing is what holds the world's Internet infrastructure together.

To build Docker we have re-used large quantities of plumbing: Linux, Go, lxc, aufs, lvm, iptables, virtualbox, vxlan, mesos, etcd, consul, systemd... the list goes on. Docker wouldn't be possible without the thousands of people who contributed to create this plumbing.When plumbing was not available or not sufficient, with the help of the Docker community, we built some of our own too. And as the volume and scope of contributions grew, so did the quality and quantity of the underlying plumbing. Of the tens of thousands of lines of code that

URL
https://www.docker.com/blog/runc/
Timestamp
Tue Jul 15 2025 14:28:50 GMT-0500 (Central Daylight Time)

and under-appreciated especially given that plumbing is what holds the world's Internet infrastructure together.

Post Categories

Community

To build Docker we have re-used large quantities of plumbing: Linux, Go, lxc, aufs, lvm, iptables, virtualbox, vxlan, mesos, etcd, consul, systemd… the list goes on. Docker wouldn't be possible without the thousands of people who contributed to create this plumbing.When plumbing was not available or not sufficient, with the help of the Docker community, we built some of our own too. And as the volume and scope of contributions grew, so did the quality and quantity of the underlying plumbing. Of the tens of thousands of lines of code that constitute the Docker platform, roughly 50% is plumbing! Docker has plumbing for interacting with both Linux and Windows native capabilities; it has plumbing for networking; service discovery; master election; security; and more.

Infrastructure Plumbing Manifesto

How we create and reuse infrastructure plumbing is fundamental to the Docker project. Our approach boils down to 3 fundamental principles which we call "the Infrastructure Plumbing Manifesto":

Whenever possible, re-use existing plumbing and contribute improvements back.

When you need to create new plumbing, make it easy to re-use and contribute improvements back. This grows the common pool of available components, and everyone benefits.

Follow the unix principles: several simple components are better than a single, complicated one.

Define standard interfaces which can be used to combine many simple components into a more sophisticated system.

There has been lots of demand for separating this plumbing from the Docker platform, so that it can be re-used by other infrastructure plumbers in accordance with infrastructure plumbing best practices. Today we are excited to announce that we are doing just that!

Spinning Out ALL Docker Infrastructure Plumbing

Starting today we will begin spinning out ALL INFRASTRUCTURE PLUMBING from the Docker best practices. Today we are excited to announce that we are doing just that!

Spinning Out ALL Docker Infrastructure Plumbing

Starting today we will begin spinning out ALL INFRASTRUCTURE PLUMBING from the Docker platform, This is a big deal, and the most important architectural change for the Docker project since its introduction.This has many benefits for Docker:

- If you are deploying Docker in production: this makes Docker more ops-friendly. Because its underlying plumbing will be more cleanly separated, the platform becomes more modular; which in turn makes it *easier to scale*, *easier to troubleshoot*, *easier to secure* and *easier to customize*.

- If you want to integrate Docker with your favorite tool: because all plumbing exposes standard interfaces, each component becomes a potential integration point.

Starting today at DockerCon, we have started to announce the first plumbing components to be spun out. But there is still plenty of work to do. WE ARE CALLING TO ALL DOCKER CONTRIBUTORS, AND ALL INFRASTRUCTURE PLUMBERS EVERYWHERE, TO JOIN THE EFFORT AND HELP US CONTRIBUTE BACK DOCKER'S PLUMBING.

Today we are introducing our most famous piece of plumbing: our OS container runtime.

Introducing runC: The universal container runtime

Docker is a platform to build, ship and run distributed applications – meaning that it runs applications in a distributed fashion across many machines, often with a variety of hardware and OS configurations. For this to be possible, it needs a sandboxing environment capable of abstracting the specifics of the underlying host (for portability), without requiring a complete rewrite of the application (for ubiquity), and without introducing excessive performance overhead (for scale).

Over the last 5 years Linux has gradually gained a collection of features which make this kind of abstraction possible. Windows, with its upcoming version 10, is adding similar features as well. Those individual features have esoteric names like "control groups", "namespaces", "seccomp", "capabilities", "apparmor" and so on. But collectively, they are known as "OS containers" or sometimes "lightweight virtualization"

Over the last 5 years Linux has gradually gained a collection of features which make this kind of abstraction possible. Windows, with its upcoming version 10, is adding similar features as well. Those individual features have esoteric names like "control groups", "namespaces", "seccomp", "capabilities", "apparmor" and so on. But collectively, they are known as "OS containers" or sometimes "lightweight virtualization".

Docker makes heavy use of these features and has become famous for it. Because "containers" are actually an array of complicated, sometimes arcane system features, we have integrated them into a unified low-level component which we simply call **runC**. And today we are spinning out runC as a standalone tool, to be used as plumbing by infrastructure plumbers everywhere.

runC is a lightweight, portable container runtime. It includes all of the plumbing code used by Docker to interact with system features related to containers. It is designed with the following principles in mind:

- Designed for security.
- Usable at large scale, in production, today.
- No dependency on the rest of the Docker platform: just the container runtime and nothing else.

Popular runC features include:

- Full support for Linux namespaces, including user namespaces
- Native support for all security features available in Linux: Selinux, Apparmor, seccomp, control groups, capability drop, pivot_root, uid/gid dropping etc. If Linux can do it, runC can do it.
- Native support for live migration, with the help of the CRIU team at Parallels
- Native support of Windows 10 containers is being contributed directly by Microsoft engineers
- Planned native support for Arm, Power, Sparc with direct participation and support from Arm, Intel, Qualcomm, IBM, and the entire hardware manufacturers ecosystem.

URL
https://www.docker.com/blog/runc/
Timestamp
Tue Jul 15 2025 14:28:50 GMT-0500 (Central Daylight Time)

- Native support of Windows 10 containers is being contributed directly by Microsoft engineers
- Planned native support for Arm, Power, Sparc with direct participation and support from Arm, Intel, Qualcomm, IBM, and the entire hardware manufacturers ecosystem.
- Planned native support for bleeding edge hardware features – DPDK, sr-iov, tpm, secure enclave, etc.
- Portable performance profiles, contributed by Google engineers based on their experience deploying containers in production.
- A formally specified configuration format, governed by the Open Container Project under the auspices of the Linux Foundation. In other words: it's a real standard.

The goal of runC is to make standard containers available everywhere

In fact, we have decided to donate the code of runC itself to the OCP foundation. Because OCP is designed to work just like the Linux Foundation, we expect that the maintainers – a blend of employees from various container-focused companies and hobbyists – will be largely left alone, and will continue to write the most awesome software possible.

runC is available today and is already under active development. Because it is based on the battle-tested plumbing used by Docker, you can use it in production today, either as part of a Docker deployment or in your own custom platform. We look forward to your contributions!

runC is available today at https://github.com/opencontainers/runc

Learn More about the Docker News from DockerCon 2015

Join our next Docker online meetup recapping all of the news from DockerCon including demos of the latest features of Docker 1.7. The meetup is on Monday, June 29 at 10:00 PDT / 19:00 CEST – click here to register!

Learn More about Docker

demos of the latest features of Docker 1.7. The meetup is on Monday, June 29 at 10:00 PDT / 19:00 CEST – click here to register!

Learn More about Docker

- Read more about the Ecosystem Technology Partner (ETP) program
- New to Docker? Try our 10 min online tutorial
- Share images, automate builds, and more with a free Docker Hub account
- Register for upcoming Docker Online Meetups
- Attend upcoming Docker Meetups
- Register for DockerCon 2015
- Start contributing to Docker

Related Posts

Docker Brings Compose to the Agent Era: Building AI Agents is Now Easy

Docker Desktop 4.43: Expanded Model Runner, Reimagined MCP Catalog, MCP Server

The Docker MCP Catalog: the Secure Way to Discover and Run MCP Servers

Docker State of App Dev: AI

The hype is real, but so are the challenges. Here's what developers, teams, and tech

Docker Brings Compose to the Agent Era: Building AI Agents is Now Easy

Define, run, and scale AI agents using Docker Compose and Docker Offload. Streamline agentic development across your stack.

Mark Cavage & Tushar Jain

Jul 10, 2025

Read now →

Docker Desktop 4.43: Expanded Model Runner, Reimagined MCP Catalog, MCP Server Submissions, and Smarter Gordon

In Docker Desktop 4.43, developers can look forward to a set of powerful updates that simplify running, managing, and securing AI models and MCP tools.

Yiwen Xu

Jul 3, 2025

Read now →

The Docker MCP Catalog: the Secure Way to Discover and Run MCP Servers

Discover why Docker is investing in the MCP ecosystem, check out our new catalog capabilities, and learn how you can help build more secure AI apps.

Nuno Coracao & Cody Rigney

Jul 1, 2025

Read now →

Docker State of App Dev: AI

The hype is real, but so are the challenges. Here's what developers, teams, and tech leaders need to know about AI's uneven, evolving role in software.

Olga Diachkova & Rebecca Floyd & Julia Wilson

Jun 25, 2025

Read now →

AI-Powered Testing: Using Docker Model Runner with Microcks for Dynamic Mock APIs

Learn how to create AI-enhanced mock APIs for testing with Docker Model Runner and Microcks. Generate dynamic, realistic test data locally for faster dev cycles.

Oleg Selajev

Jul 14, 2025

Read now →

Build a GenAI App With Java Using Spring AI and Docker Model Runner

Build a GenAI app in Java using Spring AI, Docker Model Runner, and Testcontainers. Follow this step-by-step guide to get started.

Eddú Meléndez

Jul 11, 2025

Read now →

The 2025 Docker State of Application Development Report

Explore Docker's 2025 App Dev Report: Discover trends in developer productivity, AI adoption, and security practices shaping modern software development

Olga Diachkova & Rebecca Floyd & Julia Wilson

Jul 10, 2025

Read now →

URL
https://www.docker.com/blog/runc/
Timestamp
Tue Jul 15 2025 14:28:50 GMT-0500 (Central Daylight Time)

Read now →

Read now →

Read now →

Products

- Products Overview
- Docker Desktop
- Docker Hub
- Docker Scout
- Docker Build Cloud
- Testcontainers Desktop
- Testcontainers Cloud
- Docker MCP Catalog and Toolkit
- Docker Hardened Images

Features

- Command Line Interface
- IDE Extensions
- Container Runtime
- Docker Extensions
- Trusted Open Source Content
- Secure Software Supply Chain

Developers

- Documentation
- Getting Started
- Trainings
- Extensions SDK
- Community
- Open Source
- Preview Program
- Newsletter

Pricing

- Personal
- Pro
- Team
- Business
- Pricing FAQ
- Contact Sales

Company

- About Us
- What is a Container
- Blog
- Why Docker
- Trust
- Customer Success
- Partners
- Events
- Docker System Status
- Newsroom
- Swag Store
- Brand Guidelines
- Trademark Guidelines
- Careers
- Contact Us

Languages

- English
- 日本語

URL
https://www.docker.com/blog/runc/
Timestamp
Tue Jul 15 2025 14:28:50 GMT-0500 (Central Daylight Time)